

Trabajo Práctico 12 - Despliegue de aplicaciones

1- Objetivos de Aprendizaje

- Adquirir conocimientos acerca de las herramientas de despliegue y releases de aplicaciones.
- Configurar este tipo de herramientas.

2- Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad N°: 3 (Libro Continuous Delivery: Cap 10)

3- Consignas a desarrollar en el trabajo práctico:

- Los despliegues (deployments) de aplicaciones se pueden realizar en diferentes tipos de entornos
 - On-Premise (internos) es decir en servidores propios.
 - Nubes Públicas, ejemplo AWS, Azure, Gcloud, etc.
 - Plataformas como servicios (PaaS), ejemplo Heroku, Google App Engine, etc
- Para este práctico utilizaremos como ejemplo a Heroku

4- Desarrollo:

1- Configurando Heroku

- Crear una cuenta en Heroku <https://dashboard.heroku.com>
- Instalar la utilidad de línea de comando de Heroku:
<https://devcenter.heroku.com/articles/heroku-cli>
- Abrir una línea de comandos y registrarse con la aplicación CLI

```
heroku login  
heroku container:login
```

2- Creando y Desplegando la aplicación Payroll

- Modificar el archivo Dockerfile de nuestra aplicación para que sea compatible con Heroku (necesitamos definir una variable de entorno para el puerto donde correrá el servicio):

```
FROM java:8-jre-alpine
```

```
RUN apk add --no-cache bash
```

```
WORKDIR /app
```

```
COPY target/*.jar ./spring-boot-application.jar
```

```
ENV PORT=8080
```

```
EXPOSE 8080
```

```
CMD ["java", "-Xms32m", "-Xmx128m", "-jar", "-Dserver.port=${PORT}", "-Djava.security.egd=file:/dev/./urandom", "spring-boot-application.jar"]
```

- Abrir una línea de comandos y cambiar el directorio a donde se encuentra nuestra aplicación

```
cd ./proyectos/spring-boot
```

- Crear una nueva aplicación en Heroku

```
heroku create
```

- Esto creará una aplicación con un nombre determinado, por ejemplo **ancient-reaches-06178**
- Generar y subir la imagen de Docker al registry de Heroku, desde este registry se desplegará la aplicación en Heroku

```
heroku container:push web --app=ancient-reaches-06178
```

- Una vez terminada la operación, procedemos a desplegar la aplicación

```
heroku container:release web --app=ancient-reaches-06178
```

- Nuestra aplicación estará ahora disponible en <https://ancient-reaches-06178.herokuapp.com/>

```
$ curl https://ancient-reaches-06178.herokuapp.com/employees  
{"message":"Spring boot says hello from a Docker container"}  
$
```

- Con esto vemos que está retornando el mensaje esperado.

3- Integrar el despliegue en Jenkins

- Agregar un Job o un Stage para desplegar la aplicación en Heroku
- Ejecutar los tests de Integración / UAT desde Jenkins y coleccionar los resultados utilizando esta instancia de la aplicación.